REMARKS

Reconsideration and allowance are respectfully requested.

The spelling of synchronizing and synchronization has been changed from the English style to the U.S. style per the Examiner's request in the title, abstract, and claims. The claim dependencies for claims 24 and 41 are corrected. Withdrawal of the objections raised in paragraphs 4-7 is requested.

Most of the claims stand rejected under 35 U.S.C. §103 for obviousness based on USP 6,477,638 to Gearty and newly-cited 6,381,692 to Martin. This rejection is respectfully traversed.

Gearty describes a tightly-coupled system, similar to that described in the background of the present application, where a coprocessor pipeline is synchronized with the main processor pipeline by passing signals with fixed timing from one pipeline to the other. A problem with tightly-coupled schemes is that as the length of pipeline processors increases, it is more difficult to maintain pipeline synchronization because of signal propagation delays that make it difficult to ensure signals are passed between the pipelines with the required fixed timing. Independent claims 1 and 29 solve this problem using a token-based pipeline synchronization technique where at least one synchronizing queue couples a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines. The predetermined pipeline stage and the partner pipeline stage transfer tokens to achieve a loosely-coupled synchronization scheme.

As admitted by the Examiner, Gearty's tightly-coupled scheme lacks at least three features recited in the independent claims 1 and 29. The first is a synchronizing queue that includes a FIFO buffer with a predetermined plurality of entries. The presence of multiple

entries in the queue results in variable timing in the transfer of tokens between the predetermined pipeline stage and the partner pipeline stage. If at the time a token is placed in the queue, there are no further entries ahead of the token in the queue, then that token is transferred more quickly than if at the time the token is placed in the queue when there are multiple tokens ahead of that token in the queue. The second admitted missing feature is that the token includes a tag which uniquely identifies the coprocessor instruction to which the token relates. Because there is no fixed timing for the transfer of information from the predetermined pipeline stage to the partner pipeline stage, the token uniquely identifies the coprocessor instruction to which the token relates so that the partner pipeline stage can react appropriately. The third missing feature is that synchronization is achieved "without passing signals with fixed timing between the pipelines."

The loosely-coupled synchronization scheme recited in the independent claims provides a more flexible scheme that can be used in situations where the tightly-coupled synchronization scheme may be inoperable, such as in systems where the length of the pipelined processors is large, where signal delay propagation makes it difficult to use a tightly coupled synchronization scheme, etc. Although the amount of slip between the pipelines is allowed to vary, the flexible loosely-coupled scheme ensures that the pipelines are correctly synchronized for crucial transfers of information.

Unlike the claims which are directed to a system with both a main processor and a coprocessor, Martin relates to a single asynchronous processor with plural execution units. This difference is important because the synchronizing queue in the claims couples a predetermined pipeline stage in one of the pipelines (either the pipeline of the main processor or the coprocessor) with a partner pipeline stage in the other of the pipelines (either the coprocessor or

the main processor). Given that Martin does not describe a coprocessor, it is clear that Martin cannot disclose the claimed synchronizing queue.

The Examiner identifies in Figure 1 of Martin two FIFO structures 160a and 160b. But the independent claims are not directed just to a FIFO structure. Instead, claims 1 and 29 recited a main processor with a first pipeline, a coprocessor with a second pipeline, and a synchronizing queue coupling a predetermined pipeline stage in one of the pipelines with a partner pipeline stage in the other of the pipelines. It is this synchronizing queue which includes a FIFO buffer having a predetermined plurality of entries. With only a main processor, Martin fails to teach the claimed a synchronizing queue formed using a FIFO buffer.

The FIFO 160a is coupled between the decoder and the write back unit to store and transfer ordering information to the write back unit (see column 2, lines 34 to 36), and the FIFO queue 160b is disposed between the program counter unit and the write back unit to store and transfer a program counter signal to the write back unit (see column 2, lines 39 to 41). Neither FIFO couples a predetermined pipeline stage in the main processor with a partner pipeline stage in the coprocessor or a predetermined pipeline stage in the coprocessor with a partner pipeline stage in the main processor.

The Examiner refers to column 9, lines 14 to 29, which describe the second FIFO queue 160b. As is clear from this text, the need for the FIFO queues 160b (and FIFO queue 160a) stems from the fact that Martin's processor is *asynchronous*, i.e., a processor where each functional block of the processor only operates when data arrives, rather than the various functional blocks all being operated in accordance with a system clock signal. But this has nothing to do with synchronization between a main processor and a coprocessor as claimed in this case.

The PC unit 102 generates program counter values which are used by the fetch logic to determine which instructions to fetch for execution within the relevant execution unit of the processor. A PC unit is a standard part of any main processor and does not itself execute any instructions, but instead merely determines which instructions should be fetched from memory for execution (see column 4, lines 35 to 40).

Accordingly, nothing in Martin directs a person of ordinary skill in the art to replace the latches between Gearty's main processor and coprocessor with Martin's FIFO structures.

Regarding the missing "tag which uniquely identifies the coprocessor instructions to which the token is related," the Examiner refers to column 6, lines 11 to 23 of Martin. This text merely refers to the fetching of instructions from the instruction cache by the fetch unit, and in particular, to the standard comparison of a tag portion of a fetch address with the tag entries in the instruction cache to determine whether the requested instruction is or is not within the instruction cache, i.e., whether a hit or miss condition exists. The instruction cache tags have nothing to do with placing a token in a synchronizing queue between the main processor and the coprocessor, or vice versa, where the claimed token includes a *tag* which *uniquely identifies the coprocessor instruction to which the token relates*. A lookup operation performed within an instruction cache to determine whether an instruction is present in the cache is entirely unrelated to placing a token in a synchronizing queue between a processor and a coprocessor.

Regarding the third admitted missing feature that synchronization between the first and second pipelines occurs without passing signals with fixed timing between the pipelines, the Examiner refers to column 1, lines 41 to 66 of Martin. Here, Martin provides general background information on asynchronous processors. But the present claims are not concerned with the operation of an asynchronous processor, but instead with achieving synchronization

between two separate processors: a main processor and a coprocessor used to execute certain coprocessor instructions appearing in the sequence of instructions executed by the main processor. This is clear from the language of claim 1 which states that by using at least one synchronizing queue including a FIFO buffer having a predetermined plurality of entries, and by placing tokens in that queue that include a tag which uniquely identifies the coprocessor instruction to which the token relates, the first and second pipelines (one being in the main processor and the other being in the coprocessor) are synchronised between the predetermined pipeline stage and the partner pipeline stage without passing signals with fixed timing between the pipelines. There is no teaching in Martin of passing any signals between pipelines in different processors, whether with fixed timing or otherwise.

Nor does the Examiner's rationale for combining Gearty and Martin make sense. The Examiner states that it would be obvious to "include Martin's asynchronous processor's FIFO and tag into Gearty's data processing apparatus for the benefit of implementing asynchronous processor having simpler architecture and faster processing speed." But that motivation only makes sense in Martin alone. Martin wants asynchronous operation—not synchronized. Gearty's objective is synchronized operation. The modification the Examiner is proposing completely undermines Gearty's synchronization objective. The Federal Circuit has made clear that a proposed modification which renders a prior art reference inoperable for its intended purpose is inappropriate for an obviousness inquiry. *In re Gordon*, 733 F.2d 900, 902 (Fed. Cir. 1992).

Even if one or both of Gearty's CPU and FPU each incorporated Martin's FIFOs, those FIFOs would not produce synchronization between the first and second pipelines without passing signals with fixed timing between the pipelines. Martin's FIFOs are internal to one

asynchronous processor, and neither is a synchronizing queue between a main and coprocessor. Using standard instruction cache tags in Gearty would not teach anything about a "tag which uniquely identifies the coprocessor instructions to which the token is related." So in addition to being an improper combination, the rejection based on Gearty and Martin still fails to teach at least three of the features recited in the independent claims.

The application is in condition for allowance. An early notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:sd
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100